

WHAT IS CLAIMED IS:

1. A Cloud managed class, comprising a scope field, a ScopeID field, a state field, a CloudName field, and an IsCloudNameLocal field.
2. The Cloud managed class of claim 1, wherein the state field contains a CloudState enumeration selected from the group consisting of uninitialized, synchronizing, active, invalid, disabled, stand alone, and connection lost.
3. The Cloud managed class of claim 1, further comprising at least one method exposed thereby selected from the group consisting of an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.
4. A CloudWatcher managed class, comprising a CloudWatcher constructor to instantiate a CloudWatcher object.
5. The CloudWatcher managed class of claim 4, further comprising a CloudChanged event raised when a cloud has changed in scope.
6. The CloudWatcher managed class of claim 5, wherein the CloudChanged even is raised when a cloud has been created, deleted, and updated.
7. The CloudWatcher managed class of claim 4, further comprising a GetGlobalCloud static method that returns a global cloud.
8. The CloudWatcher managed class of claim 4, further comprising a Get Clouds static method that utilizes a specified scope parameter to retrieve clouds having the specified scope.

9. The CloudWatcher managed class of claim 4, further comprising a Get Clouds static method that retrieves all clouds.

10. A PnrpEndPoint managed class, comprising a PeerName field and an IPEndPoint field.

11. The PnrpEndPoint managed class of claim 10, further comprising a first PnrpEndPoint constructor that creates a peer name that can be used for registration in a cloud.

12. The PnrpEndPoint managed class of claim 10, further comprising a second PnrpEndPoint constructor that utilizes a PeerName parameter, an IPEndPoint, and a cloud parameter to create a peer name that can be used for registration in a cloud.

13. The PnrpEndPoint managed class of claim 10, further comprising at least one method exposed thereby selected from the group consisting of an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.

14. A PnrpEndPointRegistration managed class, comprising a PeerName field, an Identity field, a RegistrationState field, a cloud field, and a SynchronizingObject field.

15. The PnrpEndPointRegistration managed class of claim 14, further comprising a first PnrpEndPointRegistration constructor that constructs a PnrpEndPointRegistration object.

16. The PnrpEndPointRegistration managed class of claim 14, further comprising a second PnrpEndPointRegistration constructor that utilizes a PnrpEndPoint parameter to construct a PnrpEndPointRegistration object.

17. The PnrpEndPointRegistration managed class of claim 14, further comprising a third PnrpEndPointRegistration constructor that utilizes a PnrpEndPoint parameter and an Identity parameter to construct a PnrpEndPointRegistration object.

18. The PnrpEndPointRegistration managed class of claim 14, further comprising a fourth PnrpEndPointRegistration constructor that utilizes a PnrpEndPoint parameter, an Identity parameter, and a TimeSpan parameter to construct a PnrpEndPointRegistration object.

19. The PnrpEndPointRegistration managed class of claim 14, further comprising at least one method exposed thereby selected from the group consisting of a register method, an unregister method, an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.

20. The PnrpEndPointRegistration managed class of claim 14, further comprising a RegistrationChanged event raised when a PnrpEndPointRegistration object changes state.

21. The PnrpEndPointRegistration managed class of claim 20, wherein the RegistrationChanged event is raised when a PnrpEndPointRegistration is unregistered, registered, and failed.

22. A PnrpEndPointResolver managed class, comprising a PeerName field, a Cloud field, a MaxResults field, a ResolveCriteria field, a TimeSpan field, and a SynchronizingObject field.

23. The PnrpEndPointResolver managed class of claim 22, further comprising a first PnrpEndPointResolver constructor for constructing a PnrpEndPointResolver object for name resolution.

24. The PnrpEndPointResolver managed class of claim 22, further comprising a second PnrpEndPointResolver constructor utilizing a PeerName parameter to construct a PnrpEndPointResolver object for name resolution.

25. The PnrpEndPointResolver managed class of claim 22, further comprising a third PnrpEndPointResolver constructor utilizing a PeerName parameter, a Cloud parameter, a MaxResults parameter, a TimeSpan parameter, and at least one ResolveCriteriaFlags parameter to construct a PnrpEndPointResolver object for name resolution.

26. The PnrpEndPointResolver managed class of claim 22, further comprising at least one method exposed thereby selected from the group consisting of a BeginResolution method, an EndResolution method, a resolve method, an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.

27. The PnrpEndPointResolver managed class of claim 22, further comprising a PeerNameFound event that is raised when a PnrpEndPoint is found.

28. The PnrpEndPointResolver managed class of claim 22, further comprising a ResolutionCompleted event that is raised when a when a maximum number of results is reached, when no PnrpEndPoint is found, and when a EndResolution method is called.

29. The PnrpEndPointResolver managed class of claim 22, further comprising a first Resolve static method that utilizes a PeerName parameter to return a PnrpEndPoint.

30. The PnrpEndPointResolver managed class of claim 29, wherein the first Resolve static method resolves one remote name synchronously.

31. The PnrpEndPointResolver managed class of claim 22, further comprising a second Resolve static method that utilizes a PeerName parameter and a Cloud parameter to return a PnrpEndPoint.

32. A method of monitoring by an application a Cloud in a managed framework, the method comprising the steps of:

communicating with a managed CloudWatcher object, the managed CloudWatcher object exposing a constructor for instantiating a CloudWatcher object; initiating the constructor.

33. The method of claim 32, wherein the managed CloudWatcher object further exposes static methods for returning a global cloud, for returning a first list of clouds associated with a scope parameter, and for returning a second list of all clouds, the method further comprising the steps of selecting one of the static methods, passing to the managed CloudWatcher object parameters required by the static method, and initiating the static method.

34. The method of claim 32, wherein the managed CloudWatcher object raises a CloudChanged event when a cloud has been created, deleted, and updated.

35. A method of managing by an application a PnrpEndPoint in a managed framework, the method comprising the steps of:

communicating with a managed PnrpEndPoint object, the managed PnrpEndPoint object exposing at least one constructor for creating a peer name that can be used for registration in a cloud, for creating a peer name that can be used for registration in the cloud from a PeerName parameter, an IPEndPoint parameter, and a cloud parameter;

selecting one of the constructors;

passing to the managed PnrpEndPoint object parameters required by the constructor selected; and

initiating the constructor.

36. A method of managing by an application a PnrpEndPoint in a managed framework, the method comprising the steps of:

communicating with a managed PnrpEndPointRegistration object, the managed PnrpEndPointRegistration object exposing at least one constructor for creating a PnrpEndPointRegistration object, for creating a PnrpEndPointRegistration object utilizing a PnrpEndPoint parameter, for creating a PnrpEndPointRegistration object utilizing a PnrpEndPoint parameter and an Identity parameter, and for creating a PnrpEndPointRegistration object utilizing a PnrpEndPoint parameter, an Identity parameter, and a TimeSpan parameter;

selecting one of the constructors;

passing to the managed PnrpEndPointRegistration object parameters required by the constructor selected; and

initiating the constructor.

37. The method of claim 36, wherein the PnrpEndPointRegistration object raises a RegistrationChanged event when a PnrpEndPointRegistration object changes state.

38. A method of resolving by an application a PnrpEndPoint in a managed framework, the method comprising the steps of:

communicating with a managed PnrpEndPointResolver object, the managed PnrpEndPointResolver object exposing at least one constructor for constructing a PnrpEndPointResolver object for name resolution, for constructing a PnrpEndPointResolver object utilizing a PeerName parameter, for constructing a PnrpEndPointResolver object utilizing a PeerName parameter, a Cloud parameter, a MaxResults parameter, a TimeSpan parameter, and at least one ResolveCriteriaFlags parameter;

selecting one of the constructors;

passing to the managed PnrpEndPointRegistration object parameters required by the constructor selected; and

initiating the constructor.

39. The method of claim 38, wherein the PnrpEndPointResolver object raises a PeerNameFound event when a PnrpEndPoint is found.

40. The method of claim 38, wherein the PnrpEndPointResolver object raises a ResolutionCompleted event when a maximum number of results is reached, when no PnrpEndPoint is found, and when an EndResolution method is called.

41. The method of claim 38, wherein the PnrpEndPointResolver object further comprises a plurality of static methods to return a PnrpEndPoint based on a PeerName parameter, and to return a PnrpEndPoint based on a PeerName parameter and a Cloud parameter, the method further comprising the steps of selecting one of the static methods, passing to the managed PnrpEndPointRegistration object parameters required by the static method selected, and initiating the static method.